

다이나믹 프로그래밍

최백준 choi@startlink.io

퇴사 2

<https://www.acmicpc.net/problem/15486>

- $N+1$ 일이 되는 날 퇴사를 하려고 한다 ($1 \leq N \leq 1,500,000$)
- 남은 N 일 동안 최대한 많은 상담을 하려고 한다
- 하루에 하나의 상담을 할 수 있고
- i 일에 상담을 하면, $T[i]$ 일이 걸리고 $P[i]$ 원을 번다

퇴사 2

<https://www.acmicpc.net/problem/15486>

- $D[i]$ = i 일이 되었을 때, 최대 수익

퇴사 2

<https://www.acmicpc.net/problem/15486>

- $D[i]$ = i 일이 되었을 때, 최대 수익
- i 일에 있는 상담을 하는 경우
- i 일에 있는 상담을 하지 않는 경우

퇴사 2

<https://www.acmicpc.net/problem/15486>

- $D[i]$ = i 일이 되었을 때, 최대 수익
- i 일에 있는 상담을 하는 경우
 - $i+T[i]$ 일이 된다.
 - 수익은 $P[i]$ 이 늘어난다.
- i 일에 있는 상담을 하지 않는 경우
 - $i+1$ 일이 된다.
 - 수익은 그대로이다.

퇴사 2

<https://www.acmicpc.net/problem/15486>

- $D[i]$ = i 일이 되었을 때, 최대 수익
- i 일에 있는 상담을 하는 경우: $D[i+T[i]] = \max(D[i+T[i]], D[i]+P[i])$
 - $i+T[i]$ 일이 된다.
 - 수익은 $P[i]$ 이 늘어난다.
- i 일에 있는 상담을 하지 않는 경우: $D[i+1] = \max(D[i+1], D[i])$
 - $i+1$ 일이 된다.
 - 수익은 그대로이다.

퇴사 2

<https://www.acmicpc.net/problem/15486>

- 정답은 $D[N+1]$ 에 들어있다.

퇴사 2

<https://www.acmicpc.net/problem/15486>

- 소스: <http://codeplus.codes/abe2fd410f50419e8e926dec94926fa5>

알약

<https://www.acmicpc.net/problem/4811>

- 약이 N 개 담겨있는 병이 있다
- 하루에 약을 하나씩 꺼내 먹는다
- 약은 한 조각 전체일 수도 있고, 쪼갠 반 조각일 수도 있다
- 한 조각이면 반으로 조각낸 다음, 반만 먹고 반을 병에 넣고
- 반 조각이면 먹는다
- 한 조각을 꺼낸 날은 W , 반 조각을 꺼낸 날은 H 를 적는다.
- 즉, $2N$ 일이 지나면, 길이가 $2N$ 인 문자열이 만들어진다.
- 이 때, 서로 다른 문자열의 개수를 구하는 문제

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F개, 반 조각이 H개 있을 때, 약을 먹는 방법의 수

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F개, 반 조각이 H개 있을 때, 약을 먹는 방법의 수
- 약을 먹는 경우
- 반 조각을 먹는 경우

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F 개, 반 조각이 H 개 있을 때, 약을 먹는 방법의 수
- 약을 먹는 경우: $D[F-1][H+1]$
- 반 조각을 먹는 경우: $D[F][H-1]$

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F개, 반 조각이 H개 있을 때, 약을 먹는 방법의 수
- 약을 먹는 경우: $D[F-1][H+1]$
- 반 조각을 먹는 경우: $D[F][H-1]$
- $D[F][H] = D[F-1][H+1] + D[F][H-1]$

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F개, 반 조각이 H개 있을 때, 약을 먹는 방법의 수
- 약을 먹는 경우: $D[F-1][H+1]$
- 반 조각을 먹는 경우: $D[F][H-1]$
- $D[F][H] = D[F-1][H+1] + D[F][H-1]$
- $D[F][0] = D[F-1][1]$
- $D[0][H] = 1$

알약

<https://www.acmicpc.net/problem/4811>

- 소스: <http://codeplus.codes/7a318b1026514d8ab7f47eeb69fe1229>

ABC

<https://www.acmicpc.net/problem/12969>

- N과 K가 주어졌을 때, 아래 두 조건을 만족하는 문자열 S를 아무거나 찾는 문제
- $3 \leq N \leq 30, 0 \leq K \leq N(N-1)/2$
- 문자열 S의 길이는 N이고, A, B, C로만 이루어져 있다
- $0 \leq i < j < N$ 이고, $S[i] < S[j]$ 를 만족하는 (i, j)쌍이 K개 있다

ABC

<https://www.acmicpc.net/problem/12969>

- $D[i][a][b][p]$ = 길이가 i 이고, A의 개수가 a 개, B의 개수가 b 개, $S[i] < S[j]$ 쌍이 p 개 있는 문자열이 가능한가?

ABC

<https://www.acmicpc.net/problem/12969>

- $D[i][a][b][p]$ = 길이가 i 이고, A의 개수가 a 개, B의 개수가 b 개, $S[i] < S[j]$ 쌍이 p 개 있는 문자열이 가능한가?
- i 번째 글자가 A인 경우
- i 번째 글자가 B인 경우
- i 번째 글자가 C인 경우

ABC

<https://www.acmicpc.net/problem/12969>

- $D[i][a][b][p]$ = 길이가 i 이고, A의 개수가 a 개, B의 개수가 b 개, $S[i] < S[j]$ 쌍이 p 개 있는 문자열이 가능한가?
- i 번째 글자가 A인 경우: $D[i+1][a+1][b][p]$
- i 번째 글자가 B인 경우: $D[i+1][a][b+1][p+a]$
- i 번째 글자가 C인 경우: $D[i+1][a][b][p+a+b]$

ABC

<https://www.acmicpc.net/problem/12969>

- 소스: <http://codeplus.codes/7c092810d59d42efb63b3f4fcbe6a1ca>

출근 기록

<https://www.acmicpc.net/problem/14238>

- A: 아무 때나 가능
- B: 출근을 하면 다음날 쉬어야 함
- C: 출근한 다음날과 다다음날 쉬워야 함
- 출근 기록 S의 모든 순열 중에서 올바른 출근 기록인 것 아무거나 찾는 문제
- S의 길이 ≤ 50

출근 기록

<https://www.acmicpc.net/problem/14238>

- 출근 기록에서 중요한 정보는 A의 개수(a), B의 개수(b), C의 개수 (c)
- B와 C때문에, 전날과 전전날 누가 일했는지가 필요하다.

출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2] = A, B, C$ 의 개수가 (a, b, c) 이고, 전날 일한 사람이 $p1$, 그 전날 일한 사람이 $p2$ 인 것이 가능한가?

출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2] = A, B, C$ 의 개수가 (a, b, c) 이고, 전날 일한 사람이 $p1$, 그 전날 일한 사람이 $p2$ 인 것이 가능한가?
- 오늘 일한 사람이 A인 경우
- 오늘 일한 사람이 B인 경우
- 오늘 일한 사람이 C인 경우

출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2] = A, B, C$ 의 개수가 (a, b, c) 이고, 전날 일한 사람이 $p1$, 그 전날 일한 사람이 $p2$ 인 것이 가능한가?
- 오늘 일한 사람이 A인 경우: $D[a+1][b][c][0][p1]$
- 오늘 일한 사람이 B인 경우: $D[a][b+1][c][1][p1]$
- 오늘 일한 사람이 C인 경우: $D[a][b][c+1][2][p1]$

출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2] = A, B, C$ 의 개수가 (a, b, c) 이고, 전날 일한 사람이 $p1$, 그 전날 일한 사람이 $p2$ 인 것이 가능한가?
- 오늘 일한 사람이 A인 경우: $D[a+1][b][c][0][p1]$
- 오늘 일한 사람이 B인 경우: $D[a][b+1][c][1][p1]$ ($p1 \neq 1$)
- 오늘 일한 사람이 C인 경우: $D[a][b][c+1][2][p1]$ ($p1 \neq 2 \ \&\& \ p2 \neq 2$)

출근 기록

<https://www.acmicpc.net/problem/14238>

- 소스: <http://codeplus.codes/db7f39d13f6a49cf9bfb92d8f22510e3>

출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$ = 앞으로 추가해야 하는 A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이 p1, 그 전날 일한 사람이 p2인 것이 가능한가?

출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$ = 앞으로 추가해야 하는 A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이 p1, 그 전날 일한 사람이 p2인 것이 가능한가?
- 오늘 일한 사람이 A인 경우
- 오늘 일한 사람이 B인 경우
- 오늘 일한 사람이 C인 경우

출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$ = 앞으로 추가해야 하는 A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이 p1, 그 전날 일한 사람이 p2인 것이 가능한가?
- 오늘 일한 사람이 A인 경우: $D[a-1][b][c][0][p1]$
- 오늘 일한 사람이 B인 경우: $D[a][b-1][c][1][p1]$
- 오늘 일한 사람이 C인 경우: $D[a][b][c-1][2][p1]$

출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$ = 앞으로 추가해야 하는 A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이 p1, 그 전날 일한 사람이 p2인 것이 가능한가?
- 오늘 일한 사람이 A인 경우: $D[a-1][b][c][0][p1]$
- 오늘 일한 사람이 B인 경우: $D[a][b-1][c][1][p1]$ ($p1 \neq 1$)
- 오늘 일한 사람이 C인 경우: $D[a][b][c-1][2][p1]$ ($p1 \neq 2 \ \&\& \ p2 \neq 2$)

출근 기록

<https://www.acmicpc.net/problem/14238>

- 소스: <http://codeplus.codes/7514be87ff434f1db780ebbace34fe38>

뮤탈리스크

<https://www.acmicpc.net/problem/12869>

- SCV N개가 있다. ($1 \leq N \leq 3$, $1 \leq \text{체력} \leq 60$)
- 뮤탈리스크 1개가 있다. 뮤탈리스크는 SCV를 공격한다.
- 한 번에 세 개의 SCV를 공격할 수 있다.
 - 첫 번째로 공격받는 SCV는 체력 9를 잃는다.
 - 두 번째로 공격받는 SCV는 체력 3을 잃는다.
 - 세 번째로 공격받는 SCV는 체력 1을 잃는다.
- SCV는 체력이 0 또는 그 이하가 되면 파괴되고, 한 번의 공격에서 같은 SCV를 여러 번 공격할 수 없다.
- SCV를 모두 파괴하기 위해 공격해야 하는 횟수의 최솟값을 구하는 문제

뮤탈리스크

<https://www.acmicpc.net/problem/12869>

- 한 번에 세 개의 SCV를 공격할 수 있다.
 - 첫 번째로 공격받는 SCV는 체력 9를 잃는다.
 - 두 번째로 공격받는 SCV는 체력 3을 잃는다.
 - 세 번째로 공격받는 SCV는 체력 1을 잃는다.
- 공격할 수 있는 순서는 총 6가지가 있다.
 - 1, 2, 3
 - 1, 3, 2
 - 2, 1, 3
 - 2, 3, 1
 - 3, 1, 2
 - 3, 2, 1

뮤탈리스크

35

<https://www.acmicpc.net/problem/12869>

- $D[i][j][k]$ = SCV의 체력이 i, j, k 일 때, 모두 파괴하기 위해 공격해야 하는 횟수의 최솟값

뮤탈리스크

<https://www.acmicpc.net/problem/12869>

- $D[i][j][k]$ = SCV의 체력이 i, j, k 일 때, 모두 파괴하기 위해 공격해야 하는 횟수의 최솟값
- 공격할 수 있는 순서는 총 6가지가 있다.
 - 1, 2, 3: $D[i-9][j-3][k-1]$
 - 1, 3, 2: $D[i-9][j-1][k-3]$
 - 2, 1, 3: $D[i-3][j-9][k-1]$
 - 2, 3, 1: $D[i-1][j-9][k-3]$
 - 3, 1, 2: $D[i-3][j-1][k-9]$
 - 3, 2, 1: $D[i-1][j-3][k-9]$

뮤탈리스크

<https://www.acmicpc.net/problem/12869>

- 소스: <http://codeplus.codes/5cb8fdc3a1e745378e9b925367ac8ec9>

Acka

<https://www.acmicpc.net/problem/12996>

- 총 N 개의 곡이 포함되어 있는 음반을 만들어야 한다.
- 음반에 포함되어 있는 모든 곡은 세 사람 중 적어도 한 명이 부른 곡이다.
- 두 음반 A와 B가 있을 때, 참여한 사람이 다른 곡이 존재한다면, 두 음반은 다른 음반이다.
- 음반을 만들 수 있는 방법의 수를 구하는 문제

Acka

<https://www.acmicpc.net/problem/12996>

- $D[N][a][b][c]$ = 곡의 수 N 개, 세 사람이 부른 곡의 수가 a, b, c 개 일 때, 경우의 수

Acka

40

<https://www.acmicpc.net/problem/12996>

- $D[N][a][b][c] = \sum D[N-1][a-i][b-j][c-k]$ (i, j, k는 세 사람의 참가 여부)

Acka

<https://www.acmicpc.net/problem/12996>

- $D[N][a][b][c] = \sum D[N-1][a-i][b-j][c-k]$ (i, j, k는 세 사람의 참가 여부)

Acka

<https://www.acmicpc.net/problem/12996>

- 소스: <http://codeplus.codes/17893006b56d41c9ac648726f8f69b2b>

플레이리스트

<https://www.acmicpc.net/problem/12872>

- N개의 곡을 이용해 길이가 P인 플레이리스트를 만들려고 한다.
- 같은 노래가 여러 개 있어도 된다.
 - 모든 노래를 플레이리스트에 추가해야 한다.
 - 같은 노래를 추가하려면, 플레이리스트에서 두 노래 사이에 적어도 M개의 곡이 있어야 한다.
- N, M, P가 주어졌을 때, 플레이리스트를 만들 수 있는 방법의 수를 구하는 문제
- $1 \leq N \leq 100, 0 \leq M \leq N, N \leq P \leq 100$

플레이리스트

<https://www.acmicpc.net/problem/12872>

- N개의 곡을 모두 추가해야 한다.
- 같은 곡 사이에는 적어도 M개의 곡이 있어야 한다.
- 길이가 M+1인 플레이리스트의 연속된 일부분은 모두 다른 곡으로 이루어져 있다.
- 같은 곡이 들어있는 플레이리스트는 길이가 M+2 이상이다.

플레이리스트

<https://www.acmicpc.net/problem/12872>

- 노래를 두 개의 그룹으로 나눌 수 있다.
- 그룹 X: 이미 추가한 노래
- 그룹 Y: 아직 추가하지 않은 노래

플레이리스트

<https://www.acmicpc.net/problem/12872>

- 노래를 두 개의 그룹으로 나눌 수 있다.
- 그룹 X: 이미 추가한 노래
- 그룹 Y: 아직 추가하지 않은 노래
- 플레이리스트를 만들 때: $X = 0, Y = N$
- 플레이리스트를 다 만든 후: $X = N, Y = 0$
- 다이나믹
- $D[P][X][Y]$ = P번째 곡을 선택할 것이고, X와 Y 집합의 크기가 X, Y인 경우의 수

플레이리스트

<https://www.acmicpc.net/problem/12872>

- 두 가지 경우로 나눌 수 있다.
- $Y > 0$ 인 경우
- Y 에 들어있는 아무 노래나 플레이리스트에 추가해도 된다

플레이리스트

<https://www.acmicpc.net/problem/12872>

- 두 가지 경우로 나눌 수 있다.
- $Y > 0$ 인 경우
- Y 에 들어있는 아무 노래나 플레이리스트에 추가해도 된다
- 그 노래는 이제 X 로 이동하게 된다
- $D[P+1][X+1][Y-1] * Y$

플레이리스트

<https://www.acmicpc.net/problem/12872>

- 두 가지 경우로 나눌 수 있다.
- $X > M$ 인 경우
- X 에 들어있는 $(X-M)$ 의 노래 중에 아무거나 추가하면 된다.

플레이리스트

50

<https://www.acmicpc.net/problem/12872>

- 두 가지 경우로 나눌 수 있다.
- $X > M$ 인 경우
- X 에 들어있는 $(X-M)$ 의 노래 중에 아무거나 추가하면 된다.
- $D[P+1][X][Y] * (X-M)$

플레이리스트

51

<https://www.acmicpc.net/problem/12872>

- $X+Y = N$ 이기 때문에
- 2차원 다이나믹으로 줄일 수 있다.

플레이리스트

<https://www.acmicpc.net/problem/12872>

- 노래를 두 개의 그룹으로 나눌 수 있다.
- 그룹 X: 이미 추가한 노래
- 그룹 Y: 아직 추가하지 않은 노래
- 플레이리스트를 만들 때: $X = 0, Y = N$
- 플레이리스트를 다 만든 후: $X = N, Y = 0$
- 다이나믹
- $D[P][X]$ = P번째 곡을 선택할 것이고, X와 Y 집합의 크기가 X, Y인 경우의 수 ($Y = N - X$)

플레이리스트

<https://www.acmicpc.net/problem/12872>

- 두 가지 경우로 나눌 수 있다.
- $Y > 0$ 인 경우
- Y 에 들어있는 아무 노래나 플레이리스트에 추가해도 된다
- 그 노래는 이제 X 로 이동하게 된다
- $D[P+1][X+1] * (N-X)$

플레이리스트

<https://www.acmicpc.net/problem/12872>

- 두 가지 경우로 나눌 수 있다.
- $X > M$ 인 경우
- X 에 들어있는 $(X-M)$ 의 노래 중에 아무거나 추가하면 된다.
- $D[P+1][X] * (X-M)$

플레이리스트

55

<https://www.acmicpc.net/problem/12872>

- 소스: <http://codeplus.codes/7680cb531a344eb3970282440fa0b365>

N-Rook II

<https://www.acmicpc.net/problem/1767>

- $N \times M$ 크기의 체스판에 룯 K 개를 놓는 경우의 수 (룩은 최대 1개의 룯에만 공격받을 수 있음)

N-Rook II

<https://www.acmicpc.net/problem/1767>

- $D[N][M][K]$ = $N \times M$ 크기의 체스판에 룯 K 개를 놓는 경우의 수
- N 번 행에 룯을 놓지 않은 경우: $D[N-1][M][K]$
- N 번 행에 룯을 하나 놓았고, 다른 룯에게 공격받지 않는 경우: $D[N-1][M-1][K-1] * M$
- N 번 행에 룯을 두 개 놓는 경우: $D[N-1][M-2][K-2] * M * (M-1) / 2$
- N 번 행에 룯을 하나 놓았고, 다른 룯에게 공격받는 경우: $D[N-2][M-1][K-2] * M * (N-1)$

N-Rook II

58

<https://www.acmicpc.net/problem/1767>

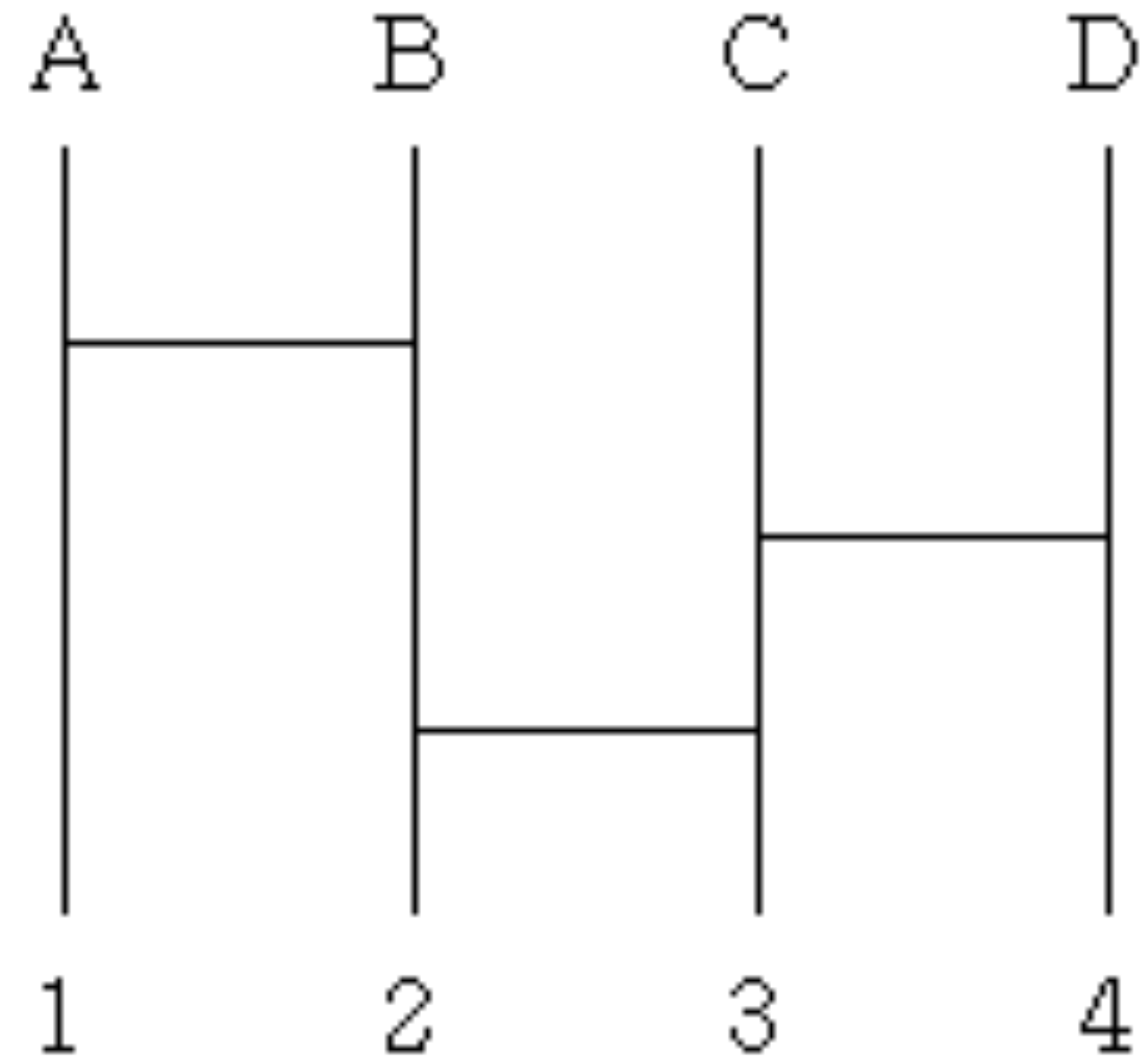
- 소스: <http://codeplus.codes/b8fcc969e932481d9e431fdc7e926b71>

사다리 게임

59

<https://www.acmicpc.net/problem/2008>

- 세로선은 N개, 가로선은 M개
- a번째 위치에서 b번째 위치로 갈 수 있게 하기 위해
- 조작하는 최소 비용
- 가로선을 지우는건 비용 X
- 그리는건 비용 Y

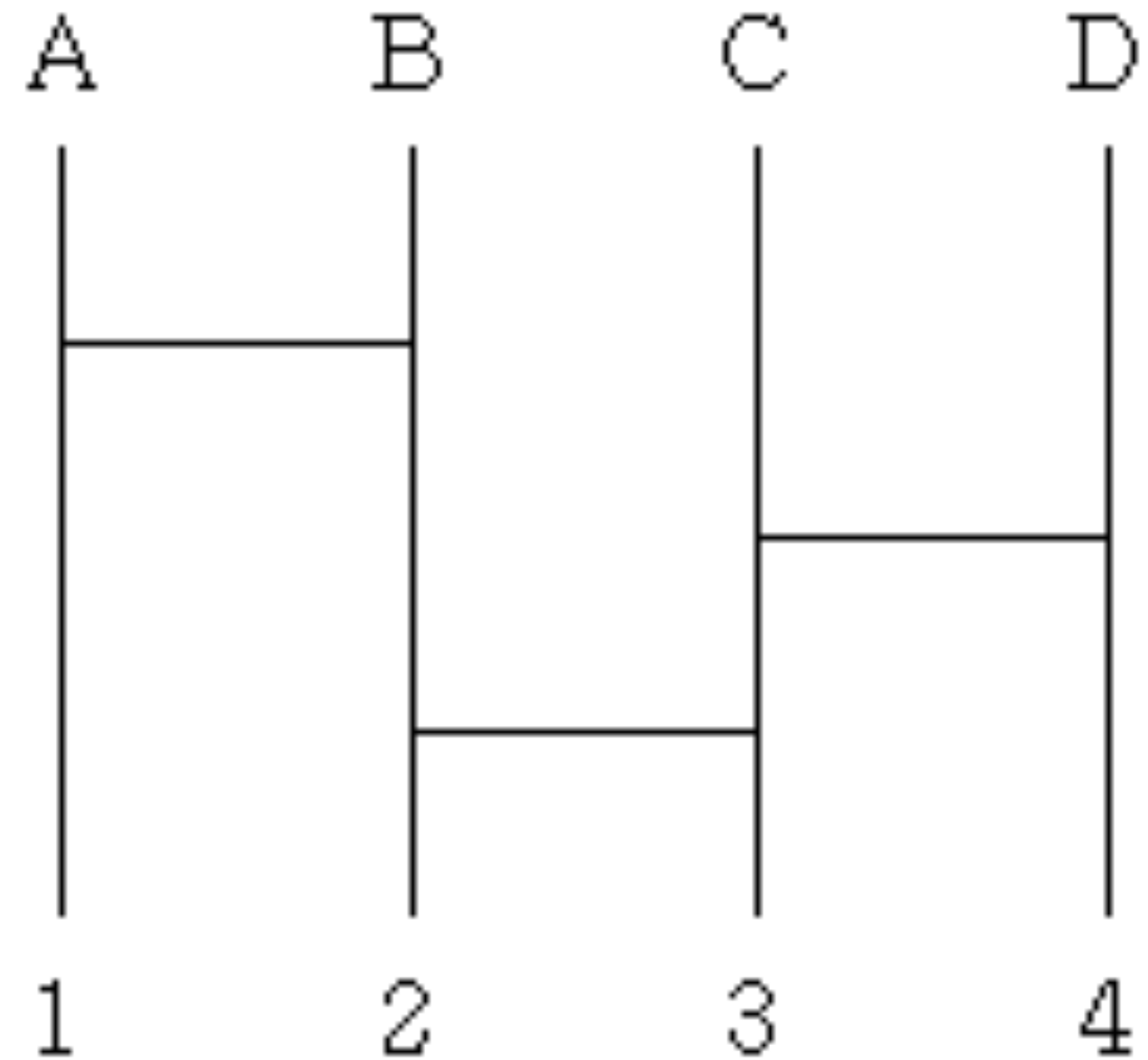


사다리 게임

60

<https://www.acmicpc.net/problem/2008>

- $D[i][j]$ = i 번째 가로선 까지 고려했을 때 start에서 j 를 가는 최소 비용



사다리 게임

<https://www.acmicpc.net/problem/2008>

```
for (int i=0; i<m; i++) {  
    if (i == start) {  
        d[0][i] = 0;  
    } else {  
        d[0][i] = abs(start - i) * add;  
    }  
}
```

사다리 게임

<https://www.acmicpc.net/problem/2008>

- 이전에 있던 위치를 k 라고 했을 때
- $j == k$ 인데, 가로선이 j 나 $j+1$ 에 위치한 경우

사다리 게임

<https://www.acmicpc.net/problem/2008>

```
if (k == j && (a[i] == k || a[i]+1 == k)) {  
    if (d[i][j] > d[i-1][k] + del) {  
        d[i][j] = d[i-1][k] + del;  
    }  
}
```

사다리 게임

<https://www.acmicpc.net/problem/2008>

- 이전에 있던 위치를 k 라고 했을 때
- 가로선이 j 와 k 사이에 위치한 경우

사다리 게임

65

<https://www.acmicpc.net/problem/2008>

```
else if ((k <= a[i] && a[i] <= j-1) || (j <= a[i] && a[i] <= k-1))
{
    if (d[i][j] > d[i-1][k] + (abs(k-j)-1)*add) {
        d[i][j] = d[i-1][k] + (abs(k-j)-1)*add;
    }
}
```

사다리 게임

<https://www.acmicpc.net/problem/2008>

- 이전에 있던 위치를 k 라고 했을 때
- 가로선이 j 와 k 사이에 위치하지 않은 경우

사다리 게임

<https://www.acmicpc.net/problem/2008>

```
else {  
    if (d[i][j] > d[i-1][k] + abs(k-j)*add) {  
        d[i][j] = d[i-1][k] + abs(k-j)*add;  
    }  
}
```

사다리 게임

<https://www.acmicpc.net/problem/2008>

- 소스: <http://codeplus.codes/e536c560b3f64ca991c2f35d8ac34f92>

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.